SUBSTITUTE SPECIFICATION

VOICE INTEGRATED VOIP SYSTEM

CROSS-REFERENCES TO RELATED APPLICATIONS

This application is related to and claims the benefit of co-pending applications No. 09/658,781, entitled "Intelligent Voice Bridging" (Atty. Docket No. 17887-007200US); No. 09/658,802, entitled "Intelligent Voice Converter" (Atty. Docket No. 17887-007300US); and No. 09/659,233, entitled "Message Store Architecture" (Atty. Docket No. 17887-007400US), all filed September 11, 2000, the disclosures of which are incorporated herein by reference.

BACKGROUND OF THE INVENTION

The present invention relates generally to the field of telecommunications application platforms or servers and more specifically to providing a gateway access server that provides telephony services and information retrieval service over a voice over IP (VOIP) network with out using any hardware cards commonly referred to as TICs (Telephony Interface Cards) and which is scalable to handle many users simultaneously.

Telecommunication application servers that provide telephony services and information retrieval service are known, however most of them use traditional PSTN (Public Switch Telephony Network) infrastructure to provide such service using various types of signaling mechanisms like T1, E1, SS7, etc.

Most recently there are some systems that provide similar service over the voice over IP (VOIP) networks. All of these systems use telephony interface cards to connect to either the PSTN or the VOIP network. An overview of a typical system is depicted in Fig. 1.

There are other systems that provide limited functionality like PC to PC and PC to phone communication services using software only model however these systems are not scalable because they perform transcode operation using the software model.

Transcoding is the process of converting one voice data format to another. All of the existing systems interact with the VOIP network using network supported CODEC format like G723.1 or G729 etc., however they a perform transcode operation on the data to convert it into either standard PCM, Mu-LAW and/or A-LAW before the application can handle the data. The cost of a phone call on a PSTN costs about 7 to 10 cents a minute while the cost of a phone call on a VoIP network has been reduced to about 1 cent a minute. Transcoding is computationally intensive operation required to be done by a special hardware device called a TIC (Telephony Interface Cards) for scalability reasons. When transcoding is done in software the system is not scalable because the transcoding operation ties up large amounts of resources. There are also systems that perform transcoding in a batch mode in a non real-time bases, i.e. offline batch processing. However this approach does not provide instant/real-time access to information until the transcode operation is complete. In some of the systems the message store stores multiple formats of the same data, one format for the VOIP/PSTN network and another format for access through the web. However such systems are either storage intensive, CPU intensive, or non-real-time oriented and cannot scale to a very large user base nor be used to provide synchronized data between the web and the telephone network.

Web portals, such as Yahoo, the assignee of the present application, receive millions of visits per day. Accordingly, standard VoIP interfacing techniques such as TICs or software transcoding add cost and complexity to implementing telephony access to services normally provided by a web browser. As is well-known, revenue generation in e-commerce is often not linked to the services provided so the cost of providing these services must be carefully controlled. On the other hand the mobility and availability of telephones to potential visitees provides a tremendous business opportunity.

Because of the above constraints, a telecommunications application server that provides functionality's like unified messaging, voice portal access to information, and communication services must use specialized hardware such as TICs. Using specialized hardware limits the server to be developed only on a platform running operating system supported by the hardware vendor. Building such a scalable application server on a platform running an operating system like Free BSD UNIX that is not supported by the hardware vendor is not possible. Further, the cost of using TICs makes the cost of implementing such a telecommunications application server prohibitive.

From the above, it is apparent the improved systems for providing telephone access to various services now provided by the internet are needed.

SUMMARY OF THE INVENTION

According to one aspect of the invention, an improved telecommunication application server handles a wide variety of call control, messaging, and information retrieval functionality using a software only model. In one embodiment, a process is started which in turn has several threads, one for each telephony channel handled by the process. The number of threads per process is configurable, it is generally set to 24 or 30 similar to the number of channels handled by a traditional T1/E1 interface. Multiple processes may run on a single system. All the processes and threads share a large amount of shared memory that contains all of the system phrases/prompts, this minimizes the amount of delay in playing phrases.

According to another aspect of the invention, if the total number of channels i.e. simultaneous telephony subscribers becomes too great for one gateway access server to handle, the system is easily scaled by adding additional gateway access servers. Each telecommunication access server maintains its own copy of the phrases/prompt data in its shared memory. There is no need to have any communication between telecommunication access servers.

According to another aspect of the invention, data received in native VoIP format is processed without transcoding so that no hardware Telephone Interface Card (TIC) of software transcoding is required.

According to another aspect of the invention, data received from the VoIP network or to be transmitted on the VoIP network is stored in native VoIP format in the shared memory thereby increasing storage efficiency.

According to another aspect of the invention, text resources, such as email, may be accessed by telephone utilizing a text-to-speech converter (TTS) which outputs voice data in non-native VoIP format. A voice coder is utilized to transcode the output of the TTS to native VoIP format.

A further understanding of the nature and advantages of the invention herein may be realized by reference to the remaining portions of the specification and the attached drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram of a typical prior art VoIP telecommunication system;

Fig. 2 is a block diagram of a preferred embodiment of the invention;

Fig. 3 is a block diagram depicting the architecture of a preferred embodiment of the voice services platform;

Fig. 4 is a block diagram depicting the architecture of a preferred embodiment of the gateway access server;

Fig. 5 is a block diagram depicting the architecture of a preferred embodiment of the VOIP API;

Fig. 6 is a block diagram depicting the architecture of a preferred embodiment of the channel thread;

Fig. 7 is a flowchart depicting steps performed to service a request for a service;

Fig. 8 is a screen shot of a web page listing voicemails messages for a service requestor; and

Fig. 9 is a screen shot of a web page implementing an applet for listening to voicemail messages transmitted over the internet in native VoIP format.

DESCRIPTION OF THE SPECIFIC EMBODIMENTS

A preferred embodiment of the invention will now be described with reference to the MyYahoo telephone interface being developed and implemented by the assignee of the present application. However, the invention is not limited to any particular implementation but has broad applicability for VOIP applications and provides many benefits which will be apparent from the following description. Users will access MyYahoo by dialing 1-800-MyYahoo from any telephone. MyYahoo will provide a universal message service including voice (such as phonemail), fax, and text (such as email). The users phone will be connected to MyYahoo servers via the internet and will use internet telephony, also known as Voice over IP (VoIP) protocols. The user requests information or services using the telephone and receives voice response generated by the MyYahoo servers.

Fig. 2 depicts the connections of an embodiment of the present invention to a Public Switched Telephone Network 200 (PSTN). Gateways 202 connect PSTN 200 to a VoIP network 204 and encode voice data in a G.723.1 format that is encapsulated in IP packets. Although embodiments of the present invention are described using a G.723.1 format, it will be understood that other formats may be used and will be appreciated by a person skilled in the art. A network interface card 206 (NIC) connects a server 208 of a VOIP system 201 to VoIP network 204. Software on server 208 processes data in the G.723.1 native VoIP format so that the need for Telephony Interface Cards (TICs) or software transcoders is eliminated. Server 208 may interact with other servers over a G.723.1 native format. As shown, system 201 may include a Message Access Server (MAS) 210 and/or Text To Speech Server (TTS) 212 in one embodiment. Also, it will be understood that system 201 is not limited to the shown servers and other components may be included and are hereinafter described. System 201 is also connected to a communication medium 214, such as the Internet.

Figure 3 shows an embodiment of a distributed client server system 300 that is used to provide telecommunication application services to callers/subscribers over a managed VOIP network 204. A preferred embodiment includes the following systems:

GAS 208 (Gateway Access Server) : GAS 208 is the primary server that is connected to VOIP network 204 over a managed IP network link. GAS 208 implements the VOIP protocol and exposes it to an application call flow using an API called VOIP_API. GAS 208 module is further described in the later part of this section. The architecture of GAS 208 is depicted in Fig. 4. As shown, GAS 208 may include any number of servers, denoted as GAS-1-GAS-N.

The Call Flow interface provides a consistent application programming interface (API) that allows internal applications, such as email readers, voice mail applications, stock quote applications, etc., to obtain the services of GAS 208 and interface with the managed VOIP network 204.

Further, a telephone applications API provides a consistent interface for third parties to write applications to obtain the services provided by GAS 208 thus additionally enhancing the scalability of the system.

MAS 302 (Message Access Server) : MAS 302 is responsible for the message store. Unlike traditional voice mail/application servers where the call flow application logic and message store are on a monolithic system, in this embodiment, the message store is separated from GAS 208, which runs the call flow and application logic. This enables the provision of a very large-scale system where GAS 208 may access any of the message stores based on the user it is currently serving. System 300 is scalable so that multiple MASs 302 may be provided.

TTS 210 (Text To Speech Server) : TTS server 210 is responsible for converting text into speech that may be played to the user. Some of the applications include providing the user with the capability of listening to email and other text based content from the phone.

ASR 304 (Automatic Speech Recognition) : ASR server 304 is responsible for recognition of voice data sent to it and translating it to text that is sent back to the requester.

VC 306 (Voice Converter) : VC 306 is a server that can convert one format of the voice into another.

WAS 308 (Web Access Server) : WAS 308 enables the subscriber to retrieve their voice and fax messages from the web. It also provides registration service and billing information access service.

AAS 310 (Add Access Server) : AAS 310 enables the call flow to have access to a set of advertisements so that it can target appropriate add for the subscriber.

NAS 312 (News Access Server) : NAS 312 stores the latest news items in a manner that can be easily accessed and played to the caller.

CAS 314 (Content Access Server) : CAS 314 provides access to content like stock quotes, weather information, sports information and customized content for the user based on My.Yahoo.com settings.

Y!Mail 316 (Yahoo Mail Servers) : GAS 208 talks to yahoo mail servers 316 to enable subscribers to listen to their email using the phone.

AB 318 (Address Book Server) : GAS 208 talks to the yahoo address book server 318 so that subscribers of this service can send messages to anyone in their address book.

UDB 320 (User Data Base Server) : UDB 320 stores the mapping between the user and MAS 302 that was allocated for that user.

The art of sending telecommunication data over managed VOIP networks is well known and will not be addressed in detail here. Essentially the user of this service will make a call to 1-800-MyYahoo. The network provider, i.e. carrier, will carry this call over their managed VOIP network 204 and will terminate the call into one of gateway access servers 208 (GAS) that is available to handle the call. GAS 208 receives an OLI (Originating Line ID), i.e. caller ID information, and may decide if it wants to answer the call or reject the call. Using the OLI information avoids any abuse of this service.

GAS 208 performs standard TCP/IP such as receiving packets, extracting data from packets received, and encapsulating data into packets to be sent.

When the user of this service dials the access number (1-800-MyYahoo), the signaling thread in a VOIP API 500 as shown in figure 5 will receive a TCP/IP signal called "call indicator" indicating that there is an incoming call. VOIP API 500 will notify the application call flow through Yahoo! Telephony API as outlined in figure 4. At this point, the application may either accept a call or reject a call. Once the application accepts the call, the signaling thread 502 will find a channel thread 504 that is ready to handle the IO and will setup a UDP connection between a channel IO thread/process 504 and VOIP network 204. All voice, fax data sent from and to the user will go through this UDP connection.

Fig. 6 is a more detailed depiction of the channel thread architecture 504 according to one embodiment. Signal processing thread 502 is called to handle channel signaling. Thread 502 detects and processes DTMF tones and CLI information signal processing thread 502 is connected to VOIP network 204 through a TCP port 608. Channel thread 504 may include a channel thread 602 and IO thread 604. IO thread 604 is connected to VOIP network 204 through a UDP port 606. IO thread 602 processes packets carrying voice data in the native VoIP format.

An embodiment of the interaction between the telecommunication access server 201 and user is depicted in the flow chart of Fig. 7. In step S700, a call is received over VOIP network 204. In step S702, the call is accepted.

Subsequent to setting up the UDP connection, the thread determines the type of service requested by the user. (step S704). Two different techniques may be implemented. The first responds to a series of DTMF tones to identify a requested service. For example, the tones generated by pressing "E" (3) followed by "M" (6) could be interpreted to be a request for email services. It is also possible for the application to play a prompt "Press 2 to listen to your email" and the subscriber will indicate its interest by pressing DTMF key "2".

Alternatively, automatic speech recognition services (ASR) may be utilized to determine voice commands such as the user saying "EMAIL". In the present embodiment, ASR utilizes voice data in Pulse Code Modulation (PCM) format so that a voice coder (VC) is utilized to convert speech commands from VoIP format to PCM format. Since only commands are converted to non-native VoIP format in this embodiment, the advantage of not decoding all incoming voice data is still substantial.

The process then determines if the service requested requires saving voice data (step S706). If so, voice data is removed from the VOIP packets (step S708) and the voice data is stored in a native VOIP format (step S710). The process then proceeds to step S712.

If saving voice data is not required or the voice data was stored in step S710, response data in the native VOIP format is accessed (step S712). The response data is then encapsulated in the native VOIP format (step S714) and response packets are sent over VOIP network 204 (step S716).

Some of the technical challenges that have to be solved in designing such a system include:

1. Jitter and prompt continuation control
2. Bi-directional packet streaming

Jitter and Prompt Continuation Control:

One of the problems encountered in designing such systems is the jitter and prompt continuation control, i.e. breakup of speech because of pauses/delays in serving voice data to VOIP network 204. To address this problem each of channel threads 504 in gateway access server (GAS) 208 includes a dedicated IO thread that maintains a voice continuity buffer that holds voice data for a smooth delivery of concatenated phrases. A concatenated phrase is a voice prompt that is built from two or more individual phrases. For example "You have 10 messages" is built from three phrases "You have" + "10" + "Messages". When this phrase is played there has to be a smoothness and continuity between each of the individual phrases. Having a configurable size look in a head continuity buffer in the IO thread provides this functionality.

When the application requests IO thread to play the phrase "You Have", IO thread plays the phrase till ninety (90 ms) milliseconds before the end. It will then return back to the application and continue to play the remaining 90 ms in the background while the application requests the next play phrase operation for "10". This process repeats till the entire phrase has been played. Further to minimize the delay in accessing the voice data for the phrases, all the phrases are stored in shared memory. In one embodiment, a 100 Meg of shared memory is used to hold half a million phrases.

Bi-Directional Packet Streaming:

Each of channels can send as well as receive data from the VOIP network 204 at any given time because telecommunication applications/networks are bi-directional applications. To support this functionality, each of channels has a dedicated thread, called the IO thread, that manages all the IO. IO thread is designed to provide directional priorities for the data handling based on the application function that is requested.

While playing the phrase or a message, IO thread gives higher priority to data transmission compared to data reception. In this mode, IO thread has to send a voice packet every 30 or 60 or 90 milli-seconds. At the same time, it has to read the data from network 204. While playing voice data, IO thread will always first transmit a voice packet and then block on the select call monitoring for incoming data. If there is any incoming data it will read the data and handle it as required. The select time out is set equivalent to the time when the next voice data has to be transmitted.

While recording a message or while waiting for the data to come in on network 204, IO thread gives higher priority to data reception and does not perform any data transmit operations. In this mode, IO thread blocks in an extended duration time out that is based on the application operation requested and will collect the data as required. For example, if the application requests a message record operation for 30 seconds, then it will block, on the selected system, calls for that duration and will collect data as it comes in.

An important aspect of bi-directional packet streaming is that while playing a voice prompt priority is always given to the out-bound data and the remaining time is used to handle the incoming data. While playing a phrase the inbound voice packet is processed during the time between two out-bound voice packets.

To address the scalability issues, the voice data is handled in the network – native format, which in this case is G723.1. This eliminates any need for hardware or software transcoding operations to convert VoIP data into either PCM, Mu-Law and/or A-Law. Because there is no transcoding operation any application that has to store data such as voice mail messages, stores them in the network native format. This functionality is provided by MAS 302, which stores all of the voice data in the G723.1 format.

The economic advantage of processing and storing data in native VoIP data is significant because no dedicated hardware TICs are required for scalability. For example, a 96 port TIC presently costs about $14,000. If each server (present cost about $3,000) can host two TICs, then the cost of a 192 port setup is $31,000 for a cost per port of $161. However, for a completely software-based system, assuming $3,000 per server, the cost of a 216 port setup is $12,000 for a cost per port of $55.55. Further, by using VoIP instead of PSTN, the cost per minute of phone call is reduced from 7 to 10 cents a minute to about 1 cent per minute for a 90% savings. If a projected 500,000 minutes of phone calls are received a day, then the savings are $45,000 per day.

Traditionally, the PCM format is used for playing and storing of messages or voice data. In the preferred embodiment, messages and data are stored in VoIP format, e.g. G.723.1, which is a factor of 10 smaller than the traditional PCM format, resulting in a reduction in storage cost of 90%,

GAS 208 has several tens to hundred of thousands of phrases/prompts that may be played to the user of this service. These prompts are stored in a large shared memory in the network native format, i.e. G723.1. All of the processes and threads that run on GAS 208 will attach to the shared memory to use the voice prompts/phrases. This method of storing the phrases/prompts in the shared memory enables the application to use the phrases/prompts with out having any additional time requirements for accessing them. The shared memory can hold several hundred thousands of phrases like the system greetings, company names, city names, letters, numbers, etc. In one embodiment, a half a million phrases are stored in 100 meg of memory and the number of phrases stored in memory, called in-RAM-phrases, can easily be increased by an allocation of more memory.

This architecture eliminates any need for GAS 208 to perform a transcoding operation because GAS 208 handles all data operations in the network native CODEC format. GAS 208 uses MAS 302 to store the messages in the network native format.

For users accessing the application using the web, WAS 308 will install a signed plug-in Java applet that can play voice messages in the network native format i.e. G723.1. This makes the message store have a single message format that is small (i.e., about 6.4 Kbps encoded data compared to 64 Kbps or 128 Kbps PCM encoding). The very small encoding size not only helps the message store to be effective, it also enables GAS 208 to handle several number of simultaneous calls coming in from VOIP network 204. One of the embodiments was tested with 96 simultaneous calls being handled by the system purely in software with vast amounts of CPU cycles still left for idling indicating that even a higher number of simultaneous calls may be handled.

A browser interface 800 is depicted in Figs. 8 and 9. In Fig. 8, browser 800 displays a web page 802 listing voice mail messages 806 received by the service requestor. In Fig. 9, a signed plug-in Java applet displays controls 900 for listening the voicemail messages stored in the native VoIP format.

In one embodiment, the architecture uses some of the products provided by other vendors such as Text To Speech 210 (TTS) and Automatic Speech Recognition 304 (ASR) that operate using standard PCM/A-Law/Mu-Law voice formats. Because of this, voice coder 306 (VC) is used to perform CODEC conversion between voice formats. VC 306 uses special boards that perform voice format conversion for TTS 210 and ASR 304 resources. Using VC 306 to transcode for limited purposes is much more efficient than transcoding all VoIP data being processed by GAS 208. Analysis has determined that only a small fraction of incoming calls, e.g., about 20%, will require TTS services so that it is much more efficient to transcode only the output of TTS 210 into a VoIP format rather than convert all incoming VoIP to standard PCM/A-Law/Mu-Law voice formats. Therefore, 80% of the conversion between formats is avoided by processing voice data in native VoIP format.

The architecture also enables intelligent information access from the telephone. This intelligence is provided by extracting the integration information from the

VOIP signaling protocol that contains the CLI (Calling line ID), i.e. caller ID information, and mapping it to V & H (vertical and horizontal) coordinates and/or city name and/or zip code. This allows the user to be located on a map. The map provides city boundaries. This information is used in selecting default content selection for the user calling for this service. For example a user calling 1-800-MyYahoo from (408) 328-7829 into the system. The system extracts the caller ID information from the VOIP network and this is used to map the user location. Based on the location of the user information like weather, sports, etc. are customized. The user can override these customizations by creating a my.yahoo.com account, in which case the defaults will be replaced with the my.yahoo.com customizations/defaults. In the case where the information requested for the exact location of the user is not available, then the search will be expanded to provide nearest location for which the requested information is accessed.

Other intelligent defaults may be provided in other contexts. For example, if the user wants to go to a nearest Italian restaurant. A list of closest choices may be created and made available to the user. When a user selects a particular choice, the location of the user is used to provide driving directions to the restaurant or other places of interest. This information may also be used to provide local time zones and time of day information.

As outlined in Figure 4, GAS 208 includes a VOIP API 400, telephony API 404, and Call Flow 406 according to one embodiment. Call Flow 406 is connected to applications such as a unified messaging application 408, an information retrieval application 410, and other applications 412. The system provides a means for any external applications 402 to be integrated into it by using the YTAP (Yahoo! Telephony Application Protocol) protocol. A particular embodiment enables external applications 402 to be accessed using YTAP by providing a VXML (Voice XML) interface cover over YTAP protocol. This may be used to integrate with external web servers and applications.

Gateway access server 208 (GAS) is capable of providing different classes of service based on the user identification. The mechanism of providing different classes of service capabilities enables the system to group users based on service requirements. For example, paid users may receive extended message save durations. Additionally, the number of messages per user groups may be based on the class to which they belong.

The invention has now been described with reference to the preferred embodiment. Alternatives and substitutions will now be apparent to persons of skill in the art. For example, the embodiments utilizing the UNIX operating system are described, however other operating system including MS NT and Windows can also be used. The terms threads and processes are utilized to have the widest meaning understood by persons of skill in the art. Different VoIP encoding schemes such as G.726 or CELP encoding may be used.

In one embodiment, the existing yahoo voice services platform is located at Yahoo! premises or at one of its co-location facilities. The telecommunication application server called GAS 208 is currently connected to VOIP network 204. The connection between VOIP network 204 and GAS 208 will carry all of the voice data from the subscriber to the application server.

Further, in the embodiments described above, when the subscriber calls Yahoo! voice services, VOIP network 204 will send a notification indication to GAS 208, indicating that there is a incoming call. At this point GAS 208 will direct network 204 to answer the call. Once network 204 answers the call, it will send call complete signal to GAS 208. At this point GAS 208 will send voice prompts like "Welcome to Yahoo! etc." Once the call has been established, actual voice data will be sent to VOIP network 204 from GAS 208 and similarly any time the subscriber talks, this data will be sent from network 204 to GAS 208.

Alternatively, in other embodiments, the integrated VOIP system may work with a VOIP network provider to encapsulate the entire Yahoo! voice services architecture into VOIP network 204 and have a control protocol that will control and manage the data using YTAP (Yahoo Telephony Application Protocol).

Accordingly, it is not intended to limit the invention except as provided by the appended claims.